

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method comprising:

first causing, by a first module of a thread in a computing environment dynamically causing, a second module to be logically chained to itselfthe first module to form or expand a thread comprised of modules, the modules of the thread including at least the first and second modules, enabling the second module to be executed as part of the same threadfirst and second modules to be executed as part of the thread and to share an execution context of the thread;

second causing, by the first module, the second module dynamically and logically chained as part of the same thread to register first events of interest to the second module with an event notification service of the computing environment;

waiting, by the first module, for notification of occurrence of one or more of the first events; and

third causing, by the first module, the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events.

2. (Currently Amended) The method of claim 1, wherein the first causing further includes invoking a thread module chaining service, the thread module chaining service having a control data structure of the thread to logically associate the second module with the first module, the control data structure including a first pointer to a first set of executable instructions of the second module designed to register the first events with the event notification service, the second module comprises a first set of executable instructions designed to register the first events with said event notification service, and a first pointer to the first set of executable instructions;

the first module dynamically causing the second module to be logically chained to itself, by invoking a thread module chaining service of the computing environment; and

the method further comprises the thread module chaining service annotating a control data structure of the thread to logically associate the second module with the first module, including with the annotation, the first pointer to the first set of executable instructions of the second module.

3. (Currently Amended) The method of claim 2, wherein the second causing further includes invoking the thread module chaining service to orchestrate registration of events of interest to logically chained modules of the thread with the event notification service, the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of executable instructions.

said causing of the second module dynamically and logically chained as part of the same thread to register first events of interest to the second module with an event notification service of the computing environment comprises the first module invoking the thread module chaining service to orchestrate registration of events of interest to logically chained modules of the thread with the event notification service; and

the method further comprises the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of executable instructions.

4. (Currently Amended) The method of claim 32, wherein the control data structure further includes a second pointer to a second set of executable instructions designed to process the occurred one of the first events.

the second module further comprises a second set of executable instructions designed to process an occurred one of the first events, and a second pointer to the second set of executable instructions; and

the thread module chaining service further includes with the annotation, the second pointer to the second set of executable instructions of the second module.

5. (Currently Amended) The method of claim 4, wherein the third causing further comprises invoking the thread module chaining service to orchestrate processing of the occurred event by the logically chained modules of the thread, the thread module chaining service retrieving the second pointer to the second set of executable instructions, and causing the second set of executable instructions to be executed, using the second pointer to locate the second set of executable instructions.

said causing of the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events comprises the first module invoking the thread module chaining service to orchestrate processing of an occurred event by the logically chained modules of the thread; and

the method further comprises the thread module chaining service retrieving the second pointer to the second set of executable instructions, and causing the second set of executable instructions to be executed, using the second pointer to locate the second set of executable instructions.

6. (Currently Amended) The method of claim 52, wherein the method further comprises

detecting, by the first module, for a thread termination condition; and fourth causing, by the first module the thread to be terminated after detecting the thread termination condition.

7. (Currently Amended) The method of claim 6, wherein the fourth causing further comprises invoking the thread module chaining service to orchestrate thread termination clean up by the logically chained modules of the thread, the thread module chaining service retrieving the a third pointer, stored in the control data structure, to a third set of executable instructions of the second module, the third set of executable instructions designed to perform termination clean up for the second module, and causing the third set of executable instructions to be executed, using the third pointer to locate the third set of executable instructions.

the second module further comprises a third set of executable instructions designed to perform termination clean-up for the second module, and a third pointer to the third set of executable instructions;

the thread module chaining service further includes with the annotation, the third pointer to the third set of executable instructions of the second module;

said detecting comprises the first module detecting for the thread termination condition;

said causing of the thread to be terminated after detecting the thread termination condition comprises the first module invoking the thread module chaining service to orchestrate thread termination clean up by the logically chained modules of the thread; and

the method further comprises the thread module chaining service retrieving the third pointer to the third set of executable instructions, and causing the third set of executable instructions to be executed, using the third pointer to locate the third set of executable instructions.

8. (Currently Amended) The method of claim 1, wherein the method further comprises comprising
the first module dynamically causing a third module to be logically chained to the first and second modules, enabling the third module to be executed as part of the same thread;

causing the third module dynamically and logically chained as part of the same thread to register second events of interest to the third module with the event notification service of the computing environment; and

causing the dynamically and logically chained third module as part of the same thread to process an occurred one of the second events, said waiting for notification of occurrence further comprising waiting for notification of one or more of the second events fourth causing, by the first module, a third module to be logically chained to the first and second modules to expand the thread, enabling the first, second, and third modules to be executed as part of the thread and to share an execution context of the thread.

9. (Currently Amended) A method comprising:

first causing, by a first module of a thread in a computing environment dynamically causing, a second module to be logically chained to itself the first module to form or expand a thread comprised of modules, the modules of the thread including at least the first and second modules, enabling the first and second modules to be executed as part of the thread and to share an execution context of the thread, enabling the second module to be executed as part of the same thread;

waiting, by the first module, for notification of occurrence of one or more of first events of interest to the second module dynamically and logically chained as part of the same thread;

second causing, by the first module, the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events;

detecting, by the first module, for a thread termination condition; and

third causing, by the first module, the thread to be terminated after detecting the thread termination condition.

10. (Currently Amended) The method of claim 9, wherein said second causing further comprises invoking a thread module chaining service to orchestrate processing of the occurred event by logically chained modules of the thread, the thread module chaining service retrieving a first pointer to a first set of executable instructions of the second module from a control data structure, the first set of executable instructions designed to process the occurred one of the first events, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of executable instructions.

the second module comprises a first set of executable instructions designed to process an occurred one of the first events, and a first pointer to the first set of executable instructions;

the method further comprises a thread module chaining service annotating a control data structure of the thread to logically associate the second module with the first module, including with the annotation, the first pointer to the first set of executable instructions of the second module;

said causing of the second module dynamically and logically chained as part of the same thread to process an occurred one of the first events comprises the first module invoking the thread module chaining service to orchestrate processing of an occurred event by logically chained modules of the thread; and

the method further comprises the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the first set of executable instructions.

11. (Cancelled)

12. (Currently Amended) The method of claim 9, wherein the third causing further comprises invoking a thread module chaining service to orchestrate thread termination clean up by logically chained modules of the thread, the thread module chaining service retrieving a first pointer to a first set of executable instructions of the second module from a control data structure, the

first set of executable instructions designed to perform termination clean up for the second module, and causing the first set of executable instructions to be executed, using the first pointer to locate the second set of executable instructions.

the second module comprises a first set of executable instructions designed to perform termination clean up for the second module, and a first pointer to the first set of executable instructions;

the method further comprises a thread module chaining service annotating a control data structure of the thread to logically associate the second module with the first module, including with the annotation, the first pointer to the first set of executable instructions of the second module;

said detecting comprises the first module detecting for the thread termination condition;

said causing of the thread to be terminated after detecting the thread termination condition comprises the first module invoking the thread module chaining service to orchestrate thread termination clean up by logically chained modules of the thread; and

the method further comprises the thread module chaining service retrieving the first pointer to the first set of executable instructions, and causing the first set of executable instructions to be executed, using the first pointer to locate the second set of executable instructions.

13. (Currently Amended) The method of claim 9, wherein the method further comprisescomprising fourth causing, by the first module, a third module to be logically chained to the first and second modules to expand the thread, enabling the first, second, and third modules to be executed as part of the thread and to share an execution context of the thread.

the first module dynamically causing a third module to be logically chained to the first and second modules, enabling the third module to be executed as part of the same thread; and

causing the third module dynamically and logically chained as part of the same thread to process an occurred one of second events, said waiting for notification of occurrence of one or more of first events further comprising waiting for notification of one or more of the second events.

14. (Currently Amended) A computing device comprising:
 - a processor; and
 - a first plurality of executable instructions to be operated by the processor to provide a thread module chaining service to
 - facilitate dynamic logical chaining of a plurality of modules to execute together as parts of a single thread and to share an execution context of the single thread, and
 - maintain and annotate a thread control data structure with control data to enable said logical chaining and orchestrated execution of the modules as parts of the single thread.
 - storage medium having stored therein a first plurality of executable instructions designed to provide a thread module chaining service to facilitate dynamic logical chaining of a plurality of modules to execute together as parts of a single thread on the computing device, including an ability to maintain and annotate a thread-control-data-structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread; and
 - at least one processor coupled to the storage medium to execute the instructions.

15. (Currently Amended) The computing device of claim 14, wherein the first plurality of executable instructions are further provide the thread module chaining service ability to maintain and annotate a thread-control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread includes an ability to annotate the thread control data structure with control data about of a module to be logically chained to be a part of a the single thread, when dynamically invoked to logically chain the module to be a part of the thread.

16. (Currently Amended) The computing device of claim 15, wherein the control data includes at least a selected one of

- a pointer of the module pointing to a plurality of executable instructions of the module to register events of interest to the module with an event notification service of the computing device,
- a pointer of the module pointing to a plurality of executable instructions of the module to process an occurred event of interest to the module, and
- a pointer of the module pointing to a plurality of executable instructions of the module to perform thread termination clean up for the module, and

the ability includes an ability to extract the selected one or more pointers from the module.

17. (Currently Amended) The computing device of claim 14, wherein the first plurality of executable instructions are further provide the thread module chaining service to first plurality of executable instructions further provide the thread module chaining service with at least a selected one of

- an ability to orchestrate registration of events of interest to the logically chained modules with an event notification service by the logically chained modules,

- an ability to orchestrate processing of an occurred event of interest by one or more of the logically chained modules, and
- an ability to orchestrate thread termination clean up of the logically chained modules by the respective logically chained modules.
18. (Original) The computing device of claim 14, wherein the computing device comprises a UPNP control point.
19. (Currently Amended) A computing device comprising:
a processor;
a first module to be operated by the processor to cause a second module to be logically chained to the first module to form or expand a thread comprised of modules, the modules of the thread including at least the first and second modules, enabling the first and second modules to be executed as part of the thread and to share an execution context of the thread; and
the second module to be operated by the processor, the second module having at least a selected one of a first set of executable instructions to register events of interest to the second module, a second set of executable instructions to process an occurred one of the events of interest, and a third set of executable instructions to perform clean up during thread termination.
storage medium having stored therein a first and a second module, with the first module equipped to logically chain the second module to the first module dynamically during execution of the first module, enabling the second module to execute with the first module as a single thread, and the second module having at least a selected one of a set of executable instructions to register events of interest to the second module, a set of executable instructions to process an occurred one of the events of interest, and a set of executable instructions to perform clean-up during thread termination; and
at least one processor coupled to the storage medium to execute the instructions.

20. (Currently Amended) The computing device of claim 19, wherein the second module further includes at least a corresponding one of a first pointer, a second pointer, and a third pointer pointing to the first set, the second set, and the third set of executable instructions respectively.

21. (Original) The computing device of claim 19, wherein the computing device comprises a UPNP control point.

22. (Currently Amended) An article of manufacture comprising:

a computer readable medium; and

a plurality of executable instructions designed to implement a thread module chaining service to

facilitate dynamic logical chaining of a plurality of modules to execute together as parts of a single thread and to share an execution context of the single thread, and

maintain and annotate a thread control data structure with control data to enable said logical chaining and orchestrated execution of the modules as parts of the single thread facilitate dynamic logical chaining of a plurality of modules to execute together as parts of a single thread in a computing environment, including an ability to maintain and annotate a thread control data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread.

23. (Currently Amended) The article of claim 22, wherein the plurality of executable instructions are further designed to ability to maintain and annotate a thread-control-data structure with control data to enable said logical chaining of the modules and their orchestrated execution as parts of a single thread includes an ability to annotate the thread control data structure with control data about a module to be logically chained to be a part of a the single thread, when dynamically invoked to logically chain the module to be a the part of the single thread.

24. (Original) The article of claim 22, wherein the first plurality of executable instructions further provide the thread module chaining service with at least a selected one of

- an ability to orchestrate registration of events of interest to the logically chained modules with an event notification service by the logically chained modules,
- an ability to orchestrate processing of an occurred event of interest by one or more of the logically chained modules, and
- an ability to orchestrate thread termination clean up of the logically chained modules by the respective logically chained modules.

25.-31. (Cancelled)